椭圆曲线的点群阶及其阶的算法

摘　要：椭圆曲线的点群阶是实现椭圆曲线密码体制的基础与条件。本文针对有限域上的椭圆曲线，介绍了椭圆曲线点群阶的定义以及所满足的定理，最后比较了两种计算阶的算法——SEA算法和SST算法。

李俊芳　崔建双／文

引　言

基于椭圆曲线的密码体制与其它密码体制相比，具有自身的优势。对于椭圆曲线密码160比特长的密钥所具有的安全性与RSA或DSA中1024比特长的密钥所具有的安全性相当，并且在同等安全强度下，它能以较小的开销（所需的计算量、存储量、带宽、软件和硬件实现的规模等）和时延（加密和签字速度高）实现较高的安全性。因此，椭圆曲线密码体制已发展成为非对称椭圆曲线密码体制研究的热点问题，成为信息安全和密码学界关注的焦点之一。

椭圆曲线密码体制的原理是基于有限域上椭圆曲线离散对数问题（ECDLP），是一个困难的问题。ECDLP的计算复杂度目前已知仅和点群的阶有关，ECDLP在其阶上有大的素因子时是一个难题，在只有指数复杂度解法时比整数分解问题（IFP）和有限域上的离散对数问题（DLP）更难。因此，构建安全的椭圆曲线密码体制，椭圆曲线的点群阶需满足一定的要求：点群阶应该是一个大的素数或者有一个大的素因子。另外，针对所随机选取的椭圆曲线判断是否是安全的椭圆曲线，也需要计算出椭圆曲线的点群阶进行判断，如超奇异椭圆曲线不适合建立椭圆曲线密码体制，易受到椭圆曲线的攻击。因此，椭圆曲线的点群阶是实现椭圆曲线密码体制的基础与条件，研究椭圆曲线的点群阶具有很重要的意义。

椭圆曲线

椭圆曲线E是一个光滑的Weierstrass方程在$P^2(K)$中的全部解(x, y)的集合。K为域，K上的摄影平面$P^2(K)$是一些等价类的集合$\{(X: Y: Z)\}$。

E：$Y^2Z+a_1XYZ+a_3YZ^2=X^3+a_2X^2Z+a_4XZ^2+a_6Z^3$。其中：$a_i \in K$。

注：

——在椭圆曲线E上恰有一个点，称之为无穷远点，即(0: 1: 0)用O表示。

——可用非齐次坐标的形式来表示椭圆曲线的Weierstrass方程。

设$x=X/Z$，$y=Y/Z$，于是原方程转化为：$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6$　　(1)

此时，椭圆曲线E就是方程(1)在摄影平面$P^2(K)$上的全部解，外加一个无穷远点O组成的集合。

对椭圆曲线可以在不同特征值的域上进行分析，特征值不同，椭圆曲线方程的表示形式也不一样。经常采用的椭圆曲线方程为：特征值K>3时

E：$y^2=x^3+ax+b$　　(2)

式中：$a, b, c \in K$，判别式$4a^3+27b^2(\mathrm{mod}p) \neq 0$。

椭圆曲线的点群阶

1. 椭圆曲线的点群阶定义

椭圆曲线的点群阶是指所定义的椭圆曲线上的点数（我们所关心的是曲线在第一象限中的整数点数）并上无穷远点（用○表示）。在此N表示椭圆曲线的点群阶。比如，椭圆曲线方程为$y^2=x^3+x+1$，$a=b=1$；$p=13$；$4a^3+27b^2=31\mathrm{mod}p=5\mathrm{mod}(13) \neq 0$。

椭圆曲线$E_{13}(1, 1)$上的整数点为：(0, 1)，(0, 12)，(1, 4)，(1, 9)，(4, 2)，(4, 11)，(5, 1)，(5, 12)，(7, 0)，(8, 1)，(8, 12)，(10, 6)，(10, 7)，(11, 2)，(11, 11)，(12, 5)，(12, 8)，共17个点，再加上无穷远点O，椭圆曲线$E_{13}(1, 1)$的点群阶N为18。

2. 椭圆曲线点群阶的特点

（1）椭圆曲线的点群阶满足Hasse定理

Hasse定理：令F_q表示q个元素的有限域，用$E(F_q)$表示定义在F_q上的一个椭圆曲线E，$q=p^m$，p为素数且为F_q的特征值。$E(F_q)$的点数用$\#E(F_q)$表示，则$\#E(F_q)$满足以下关系：

$|\#E(F_q)-q-1| \leqslant 2q^{1/2}$

不同特征值的有限域的选取，椭圆曲线的方程表示形式也不同。本文所选取的有限域为F_p(素数域)，p为素数，p>3，此时，Hasse定理变为：$|\#E(F_p)-p-1| \leqslant 2p^{1/2}$，即$p+1-2p^{1/2} \leqslant \#E(F_p) \leqslant p+1+2p^{1/2}$。

举例如：设参数a=b=1，椭圆曲线的方程为$y^2=x^3+x+1$。椭圆曲线的点群阶随p值变化的结果见表1。

表1 椭圆曲线的点群阶随P值变化的结果

P	19	97	571	997
N	21	97	568	995

点数N满足Hasse定理，即$p+1-2p^{1/2} \leqslant \#E(F_p) \leqslant p+1+2p^{1/2}$。当p=19时，按Hasse定理$19+1-2*19^{1/2} \leqslant \#E(F_p) \leqslant 19+1+2*19^{1/2}$，即$11.28 \leqslant \#E(F_p) \leqslant 28.72$，所求出的N值为21，在此范围内。当P=97时，按Hasse定理$97+1-2*97^{1/2} \leqslant \#E(F_p) \leqslant 97+1+2*97^{1/2}$，即$78.30 \leqslant \#E(F_p) \leqslant 117.70$，所求出的N值为97，在此范围内。当P=997时，按Hasse定理$997+1-2*997^{1/2} \leqslant \#E(F_p) \leqslant 997+1+2*997^{1/2}$，即$934.85 \leqslant \#E(F_p) \leqslant 1061.15$，所求出的N值为995，在此范围内。

(2) 根据表1可以看出，随着p值的增大，N值也增大，即椭圆曲线上的点数增多。

3. 椭圆曲线点群阶的计算方法

判断所选取的椭圆曲线是否是非超奇异曲线，需要计算出椭圆曲线的阶。另外，利用有限域上的椭圆曲线进行加密和数字签名体制时，椭圆曲线的阶也应当是已知的。所以说椭圆曲线的阶的计算非常重要。如何快速有效地计算所选取的椭圆曲线的点群阶是研究椭圆曲线密码体制的核心问题之一。目前，有两种处理此问题的方法：复乘方法，即构造给定阶的椭圆曲线；随机选取椭圆曲线参数，计算它的阶，直到找到素数(或拟素数)阶椭圆曲线。复乘方法产生的椭圆曲线具有附带的结构特征，从安全性角度来说，这是一个潜在的威胁；而随机选取的方法是比较理想的。在这方面，主要有两种求阶方法：l-adic方法和p-adic方法。l-adic求阶方法主要指的是SEA算法，p-adic求阶方法中有Satoh算法、SST算法和AGM算法。下面主要针对SEA算法和SST算法进行介绍和比较。

(1) SEA算法

R.Schoof作了开创性的工作，提出了著名的Schoof算法，后经Elkies和Atkin的改进，该算法具有了实用价值，被称作SEA算法。SEA算法采用先求F_q上椭圆曲线群的阶模小素数l的局部信息，再综合成整体结论的思路，所以也被称为l-adic求阶方法。

下面简单地给出了SEA算法计算椭圆曲线点群阶的步骤。

随机选取素数阶椭圆曲线：

输入：有限域特征p。

输出：椭圆曲线方程$y^2=x^3+ax+b$。

①随机选取椭圆曲线参数a，b $\in$ Fp，$a \times b \neq 0$，$\triangle=4a^3+27b^2 \pmod p \neq 0$。

②对Atkin素数，计算tmodl的信息；对Elkies素数，判断出子多项式h(x)在F_p中是否有根，若有根x_p且$(x^3_p+ax_p+b)/p=1$，则返回①。否则计算tmodl，若p+1=tmodl，返回①。

③计算$\#E(F_p)$并进行素性测试，若$\#E(F_p)$为合数，返回①。否则，输出椭圆曲线方程$y^2=x^3+ax+b$。

(2) SST算法

SST算法求阶的思路是把F_q上椭圆曲线提升到p-adic域Q_q上，再求阶。通常包括三步。定义E上的小Frobenius映射$\sigma:(x,y)\rightarrow(x^p,y^p)$，分别记为F，$\sigma$的对偶同种映射为$\hat{}F$，$\hat{}\sigma$。

①首先构造F_q上椭圆曲线在p-adic域Q_p上的n次非分歧扩张K上的典型提升；

②计算$\hat{}\sigma$的提升映射的核；

③求取$\hat{}F$的提升映射在提升曲线的形式群上诱导的同态的一次项的系数。

SST算法与SEA算法的区别：SST算法是针对特征为2的有限域上求椭圆曲线点群阶的算法当中目前较为有效的一种，而且在实际应用中得到了验证，其时间复杂度为$O(lb^4q)$，空间复杂度为$O(lb^2q)$。但是SST算法需要预运算，所以不能在智能卡上实现。SEA算法的时间复杂度为$O(lb^6q)$，不如SST算法。但是，由于SST算法计算复杂度前的常数和基域的特征p有密切的关系，所以当基域的特征p较大时，SEA算法比SST算法有效。

结 论

椭圆曲线的点群阶是椭圆曲线密码体制当中一个重要的概念，也是建立安全椭圆曲线密码体制一个不可缺少的条件。本文详细地解释了椭圆曲线的点群阶的定义，并举例论述了点群阶的特点，最后介绍了两种计算点群阶的算法——SST算法与SEA算法，并进行了比较，对研究椭圆曲线密码体制以及建立椭圆曲线密码体制具有参考意义。

教育部科学技术研究重点项目资助(项目编号:01021)；北京科技大学研究生研究基金资助。